

**Dong J, Han Z-F, Zhao Y, Wang W, Prochazka A, Chambers JA.**  
**Sparse Analysis Model Based Multiplicative Noise Removal with Enhanced**  
**Regularization.**

*Signal Processing* 2017

DOI: <http://dx.doi.org/10.1016/j.sigpro.2017.01.032>

**Copyright:**

© 2017. This manuscript version is made available under the [CC-BY-NC-ND 4.0 license](#)

**DOI link to article:**

<http://dx.doi.org/10.1016/j.sigpro.2017.01.032>

**Date deposited:**

09/02/2017

**Embargo release date:**

03 February 2018



This work is licensed under a  
[Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence](#)

# Sparse Analysis Model Based Multiplicative Noise Removal with Enhanced Regularization<sup>☆</sup>

Jing Dong<sup>1,\*</sup>, Zifa Han<sup>2</sup>, Yuxin Zhao<sup>3</sup>, Wenwu Wang<sup>4</sup>, Ales Prochazka<sup>5</sup>,  
Jonathon Chambers<sup>6</sup>

---

## Abstract

The multiplicative noise removal problem for a corrupted image has recently been considered under the framework of regularization based approaches, where the regularizations are typically defined on sparse dictionaries and/or total variation (TV). This framework was demonstrated to be effective. However, the sparse regularizers used so far are based overwhelmingly on the synthesis model, and the TV based regularizer may induce the stair-casing effect in the reconstructed image. In this paper, we propose a new method using a sparse analysis model. Our formulation contains a data fidelity term derived from the distribution of the noise and two regularizers. One regularizer employs a learned analysis dictionary, and the other regularizer is an enhanced TV by introducing a parameter to control the smoothness constraint defined on pixel-wise differences. To address the resulting optimization problem, we adapt the alternating

---

<sup>☆</sup>This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/K014307/1 and the MOD University Defence Research Collaboration in Signal Processing.

\*Corresponding author.

<sup>1</sup>J. Dong is with the College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, Jiangsu, China. (email: jingdong@njtech.edu.cn).

<sup>2</sup>Z.-F. Han is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. (email: zifahan@gmail.com).

<sup>3</sup>Y. Zhao is with the College of Automation, Harbin Engineering University, Harbin, Heilongjiang, China. (email: zhaoyuxin@hrbeu.edu.cn).

<sup>4</sup>W. Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (email: w.wang@surrey.ac.uk).

<sup>5</sup>A. Prochazka is with the University of Chemistry and Technology, Prague, Czech Republic. (email: a.prochazka@ieec.org).

<sup>6</sup>J. Chambers is with the School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K. (email: Jonathon.Chambers@newcastle.ac.uk).

direction method of multipliers (ADMM) framework, and present a new method where a relaxation technique is developed to update the variables flexibly with either image patches or the whole image, as required by the learned dictionary and the enhanced TV regularizers, respectively. Experimental results demonstrate the improved performance of the proposed method as compared with several recent baseline methods, especially for relatively high noise levels.

*Keywords:* Multiplicative noise, analysis sparse model, dictionary learning, smoothness regularizer

---

## 1. Introduction

Multiplicative noise, also known as speckle noise, is often observed in synthetic aperture radar (SAR) and sonar (SAS) images, due to the effect of interference introduced in their acquisition processes [1]. Compared to additive Gaussian noise often assumed in traditional image denoising, removing speckle noise is deemed to be more difficult for two reasons. Firstly, the noise is multiplied with (rather than added to) the original image, which usually degrades the images more severely as compared with additive noise [2]. Secondly, the study of the statistical properties of speckle noise indicates that Gamma and Rayleigh distributions are more suitable for modelling such noise [1], [2], [3], [4] instead of the widely used Gaussian distribution in conventional image denoising, and thus the data fidelity term derived from the noise model is not quadratic, raising difficulties for optimization.

Mathematically, the observed image  $\mathbf{w} \in \mathbb{R}^N$  (reshaped from a  $\sqrt{N} \times \sqrt{N}$  image) contaminated by the speckle noise  $\mathbf{u} \in \mathbb{R}^N$ , can be represented as [4], [5]

$$\mathbf{w} = \mathbf{g} \circ \mathbf{u}, \quad (1)$$

where  $\mathbf{g} \in \mathbb{R}^N$  denotes the image to be restored. The symbol  $\circ$  denotes the Hadamard product (i.e. entry-wise product) of two matrices/vectors. The aim of despeckling is to estimate  $\mathbf{g}$  from the observed image  $\mathbf{w}$ . In this paper, we focus on Gamma distributed multiplicative noise, such that the elements of  $\mathbf{u}$  are

assumed to be independent and identically distributed (i.i.d.) with probability density function given by [2], [4], [5]

$$f_u(u) = \frac{L^L}{\Gamma(L)} u^{L-1} e^{-Lu}, \quad (2)$$

where  $L$  is a positive integer defining the noise level and  $\Gamma(\cdot)$  is the classical  
 15 Gamma function given by  $\Gamma(L) = (L-1)!$ . A smaller  $L$  indicates stronger noise.

### 1.1. Related Work

Classical methods for removing multiplicative noise are spatial filtering [6],  
 [7], [8] and wavelet domain filtering [9], [10]. More recently, regularization based  
 20 approaches to denoising, where the image reconstruction task is formulated as  
 an optimization problem with regularizers, have attracted much attention [4],  
 [5], [11], [12], [13]. A popular regularizer employed in these approaches is total  
 variation (TV) which was proposed originally for reducing additive Gaussian  
 noise [14]. The TV-based methods were then used for multiplicative noise in  
 25 the original image domain as in equation (1) or in the log-domain by applying a  
 logarithmic transform. Typical examples performed in the original domain are  
 the first TV-based multiplicative noise removal method proposed in [15] and the  
 method of Aubert-Aujol (AA) [11]. The method in [15] minimizes the TV of the  
 image to be recovered with the constraints exploiting the mean and variation  
 30 of the noise, but this method is not effective for removing Gamma distributed  
 noise as the noise considered in its restoration model is assumed to follow a  
 Gaussian distribution. The AA method [11] exploits a Bayesian *maximum a  
 posteriori* (MAP) estimate, yielding an image restoration model consisting of  
 a data fidelity term based on the prior distribution of the multiplicative noise  
 35 and a TV regularization term. However, the quality of the image restored  
 by the AA method may be limited by the local solutions obtained from the  
 optimization of a non-convex model. Another class of denoising methods based  
 on the TV regularizer considers the image restoration in the log-domain [4],

[5], [12], [13], aiming to simplify the multiplicative noise model as an additive  
 40 model which is easier to deal with than the original model. In general, the  
 reconstruction models employed in these methods commonly consist of a data  
 fidelity term and regularization terms reflecting prior information related to the  
 image. However, the formulations of these terms and optimization approaches  
 may differ substantially. In [12], Shi and Osher (SO) considered both the data  
 45 fidelity and TV terms of the AA method [11] in the log-domain to overcome  
 the non-convex optimization issue. Multiplicative Image Denoising with the  
 Augmented Lagrangian (MIDAL) algorithm [4] uses the same model as used by  
 SO but applies a different optimization framework based on variable splitting  
 and augmented Lagrangian for better numerical efficiency. Apart from the data  
 50 fidelity term and the TV regularization as in the reconstruction model used  
 by SO [12] and the MIDAL algorithm [4], the method presented in [13] also  
 incorporates a quadratic data fitting term to apply the TV term in a more  
 efficient manner, but it tends to be outperformed by the MIDAL algorithm [4].

Although the TV regularization proves to be effective for reducing multi-  
 55 plicative noise, the smoothly varying regions in the original image are usually  
 recovered as piecewise constant areas, which is also well known as the stair-  
 casing effect [2]. An approach to avoid this issue is to introduce priors on the  
 image to be recovered. Recently, the sparsity prior was shown to be helpful  
 for the reconstruction of images with multiplicative Gamma noise [2], [5], [16].  
 60 Duran, Fadili and Nikolova (DFN) [2] adopted the sparsity prior by consider-  
 ing the sparsity of the image in the curvelet transformed domain and restoring  
 the frame coefficients via a TV regularized formulation in the log-domain. As  
 dictionaries learned from the related data have the potential to fit the data  
 better than pre-defined dictionaries, dictionary learning techniques in sparse  
 65 representation have also been utilized to model the sparsity prior [5], [16]. The  
 methods proposed in [16] and [5] both introduce dictionary learning to the TV  
 regularized model [12], [4], but with different frameworks. These two methods  
 are referred to as MNR-DL-TV-1 (multiplicative noise removal via dictionary  
 learning and total variation) [16] and MNR-DL-TV-2 [5] respectively. In these

two methods, the dictionary is learned by the K-SVD algorithm [17] which is a well-known dictionary learning method based on the sparse synthesis model. The MNR-DL-TV-1 method performs noise reduction in two stages: the image is first denoised using the learned dictionary; and then a model based on an  $\ell_2$  data fidelity term and TV regularization is applied to further improve the denoising result. In contrast, the MNR-DL-TV-2 method formulates the image reconstruction task as an optimization problem containing two regularizers: a learned dictionary based term and a TV term. However, we have found that the performance of MNR-DL-TV-2 is limited for relatively high noise-levels, as shown in our simulations (see Section 5.1 later).

It should be noted that the learned dictionaries employed in the MNR-DL-TV-1 [16] and MNR-DL-TV-2 [5] methods are both based on the sparse synthesis model [17]. In recent years, the sparse analysis model, as a counterpart of the synthesis model, has attracted much attention [18], [19]. Dictionary learning based on the sparse analysis model was also shown to be effective in the reduction of additive Gaussian noise [20], [21], however, few researchers have studied its potential for removing multiplicative noise. We have proposed a speckle noise removal method in [22] which applies the dictionary learned based on the analysis model to the regularizer of the restoration formulation. This approach, referred to as Removing Speckle Noise via Analysis Dictionary Learning (RSN-ADL), has the ability to preserve details while reducing multiplicative noise, however the smooth regions are not well-recovered, as will be illustrated in Section 5.

### 1.2. Contributions

In this paper, we propose a new model for reconstructing the image from a multiplicative noise corrupted image and develop a novel method for optimizing this model. The proposed method applies a sparse analysis model based regularizer and a smoothness regularizer. The joint employment of these two regularizers, which is different from the existing methods, aims to exploit the benefits of both priors and partly addresses the limitations of the existing meth-

ods mentioned above. Specifically, the sparse analysis model based regularizer  
 is constructed with an analysis dictionary learned from image patches via the  
 Analysis SimCO algorithm [23], [21], and the smoothness regularizer is formed  
 based on the pixel-wise differences in the horizontal and vertical directions. This  
 reconstruction model extends our previous work [22] by introducing the smooth-  
 ness regularization term. Since the dictionaries used in the regularizer of [22]  
 are usually well adapted to textures but not for smooth areas [5], the introduc-  
 tion of the smoothness regularizer in the proposed model has the potential to  
 overcome this issue. Compared with the methods based on TV regularization,  
 for example the MIDAL algorithm [4], the proposed model can mitigate the  
 stair-casing effect appearing in the recovered images due to the application of  
 the analysis model based regularization, as will be demonstrated in Section 5.  
 The proposed model also shows advantages for a relatively high level of noise,  
 compared with the DFN [2] and MNR-DL-TV-2 [5] algorithms.

The introduction of the two regularizers in our restoration formulation, how-  
 ever, renders the optimization task non-trivial, especially since the two regular-  
 izers are defined from different representations of the image. In particular, the  
 dictionary is learned with image patches instead of the whole image in order  
 to reduce the computational complexity. As a result, the sparse analysis model  
 based regularizer is represented with image patches. The smoothness regular-  
 izer, on the other hand, is defined with pixel-wise differences calculated across  
 the whole image. In order to address the optimization of the presented model,  
 we propose a new method based on the framework of the alternating direction  
 method of multipliers (ADMM) [24]. Two auxiliary variables are introduced  
 to split the variables by reformulating our approach as a constrained optimiza-  
 tion problem, and then the ADMM framework is applied to decompose the  
 optimization as a sequence of sub-problems which are easier to solve. In the  
 sub-problem related to the smoothness regularizer, there exist two variables in  
 different forms, and thus an approximation technique is applied to relax the  
 original sub-problem as a problem with a unified variable.

### 130 1.3. Notations

Bold capital letters are used to represent matrices. The notation  $\mathbf{X}_{i,:}$  is used to specify the  $i$ -th row of the matrix  $\mathbf{X}$  and  $\mathbf{X}_{:,j}$  represents its  $j$ -th column. Bold lowercase letters represent vectors. Scalars are either capital or lowercase letters. The norms  $\|\cdot\|_1$ ,  $\|\cdot\|_2$  and  $\|\cdot\|_F$  denote the  $\ell_1$ -norm,  $\ell_2$ -norm and the  
135 Frobenius norm respectively. When the operand of  $\|\cdot\|_1$  is a matrix, it denotes the sum of the absolute values of the elements in the matrix, which is different from the canonical definition of the  $\ell_1$ -norm for matrices. The notation  $|\cdot|$  returns the absolute value of a scalar. The notation  $\langle \cdot, \cdot \rangle$  is used to represent the canonical inner-product of two vectors.

### 140 1.4. Organization of the Paper

As the dictionary used in our image restoration model is learned based on the analysis model via the Analysis SimCO algorithm, Section 2 reviews the analysis model and the Analysis SimCO algorithm briefly to make this paper self-contained. The proposed image restoration model is introduced in Section  
145 3, followed by Section 4 where the optimization method is presented. The experimental results with known test images and real SAR images corrupted by speckle noise are presented in Section 5, and Section 6 concludes the paper.

## 2. Analysis Model and Analysis SimCO Algorithm

For a signal  $\mathbf{a} \in \mathbb{R}^m$ , the sparse analysis model assumes that the product of  
150  $\mathbf{\Omega} \in \mathbb{R}^{p \times m}$  and  $\mathbf{a}$  is sparse, i.e.  $\mathbf{x} = \mathbf{\Omega}\mathbf{a}$  with  $\|\mathbf{x}\|_0 = p - l$ , where the  $\ell_0$ -norm  $\|\cdot\|_0$  counts the number of non-zero elements of its argument and  $0 \leq l \leq p$  is the co-sparsity of  $\mathbf{a}$  [18]. The matrix  $\mathbf{\Omega}$  is usually referred to as an analysis dictionary, with each row of  $\mathbf{\Omega}$  being an atom. The vector  $\mathbf{x} \in \mathbb{R}^p$  is the analysis representation of the signal  $\mathbf{a}$  with respect to  $\mathbf{\Omega}$ . In this model, the analysis  
155 dictionary  $\mathbf{\Omega}$  plays an important role, and the dictionaries learned from a set of training signals show some advantages compared with pre-defined dictionaries [20].



Given a set of training data contained in  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , the analysis dictionary learning problem can be formulated as [25]

$$\begin{aligned} \{\mathbf{\Omega}^*, \mathbf{X}^*\} &= \arg \min_{\{\mathbf{\Omega}, \mathbf{X}\}} \|\mathbf{X} - \mathbf{\Omega}\mathbf{A}\|_F^2 \\ \text{s.t. } &\|\mathbf{X}_{:,i}\|_0 = p - l, \forall i. \end{aligned} \quad (3)$$

This is a general formulation without any additional constraint on  $\mathbf{\Omega}$  apart from the co-sparsity constraints  $\|\mathbf{X}_{:,i}\|_0 = p - l, \forall i$ . However, this formulation has ambiguities caused by scaling [21]. In order to avoid these ambiguities, unit  $\ell_2$ -norm constraints on the rows of  $\mathbf{\Omega}$  are applied, leading to the following formulation of the Analysis SimCO algorithm [21], [23],

$$\begin{aligned} \{\mathbf{\Omega}^*, \mathbf{X}^*\} &= \arg \min_{\{\mathbf{\Omega}, \mathbf{X}\}} \|\mathbf{X} - \mathbf{\Omega}\mathbf{A}\|_F^2 \\ \text{s.t. } &\|\mathbf{X}_{:,i}\|_0 = p - l, \forall i \\ &\|\mathbf{\Omega}_{j,:}\|_2 = 1, \forall j. \end{aligned} \quad (4)$$

The Analysis SimCO algorithm solves the above problem by an optimization framework alternating between two stages: analysis sparse coding stage and dictionary update stage. The procedure of the Analysis SimCO algorithm is summarized in Algorithm 1 and more details are presented below.

The purpose of the analysis sparse coding stage is to obtain the sparse representation  $\mathbf{X}$  of the training signals in  $\mathbf{A}$  based on a given dictionary  $\mathbf{\Omega}$ . The exact representation  $\mathbf{X}$  can be calculated directly by simply multiplying  $\mathbf{A}$  by the dictionary  $\mathbf{\Omega}$ , that is

$$\mathbf{X} = \mathbf{\Omega}\mathbf{A}. \quad (5)$$

Since the initial dictionary is an arbitrary one, the representation obtained in this way may not satisfy the co-sparsity constraints in (4). A hard thresholding operation is therefore applied to enforce the co-sparsity

$$\hat{\mathbf{X}} = HT_l(\mathbf{X}), \quad (6)$$

---

**Algorithm 1** Analysis SimCO

---

**Input:**  $\mathbf{A}$ ,  $p$ ,  $l$

**Output:**  $\mathbf{\Omega}^*$

**Initialization:**

Initialize the iteration counter  $k = 1$  and the analysis dictionary  $\mathbf{\Omega}^{(k)}$ .  
Perform the following steps.

**Main Iterations:**

1. Analysis sparse coding: Compute the representation  $\mathbf{X}^{(k)}$  with the fixed dictionary  $\mathbf{\Omega}^{(k)}$  and the training signals in  $\mathbf{A}$ , based on equations (5) and (6).
  2. Dictionary update:
    - (a) Compute the negative gradient  $\mathbf{H}$ , based on equation (8).
    - (b) Compute the search direction  $\bar{\mathbf{h}}_j$  for  $j = 1, 2, \dots, p$ , based on equation (9).
    - (c) Update the dictionary  $\mathbf{\Omega}^{(k+1)} \leftarrow \mathbf{\Omega}^{(k)}$ , based on equation (10).
  3. If the stopping criterion is satisfied,  $\mathbf{\Omega}^* = \mathbf{\Omega}^{(k+1)}$ , quit the iteration. Otherwise, increase the iteration counter  $k = k + 1$  and go back to step 1.
- 

where  $HT_l(\mathbf{X})$  is the non-linear operator that sets the smallest  $l$  elements (in magnitude) of each column of  $\mathbf{X}$  to zeros. The representation  $\hat{\mathbf{X}}$  obtained via equation (6) is the best approximation of the exact representation  $\mathbf{X}$  in terms of the error in Frobenius norm among all the matrices satisfying the co-sparsity constraints.

In the dictionary update stage,  $\mathbf{\Omega}$  is updated assuming known and fixed  $\mathbf{X}$ . In other words, this stage aims at optimizing the following problem

$$\arg \min_{\mathbf{\Omega}} \|\mathbf{X} - \mathbf{\Omega}\mathbf{A}\|_F^2 \quad \text{s.t.} \quad \|\mathbf{\Omega}_{j,:}\|_2 = 1, \quad \forall j. \quad (7)$$

Since the Stiefel manifold  $\mathcal{S}_{m,1}$  is defined as  $\mathcal{S}_{m,1} = \{\mathbf{s} \in \mathbb{R}^m : \mathbf{s}^T \mathbf{s} = 1\}$  [26], the transpose of each row in  $\mathbf{\Omega}$  can be seen as one element in  $\mathcal{S}_{m,1}$ . Thus, one of the “line” search methods on manifolds can be utilized to deal with problem (7). In Analysis SimCO, the gradient descent method on manifolds is applied.

Specifically, given that the negative gradient of the objective function in (7)

with respect to  $\mathbf{\Omega}$  is

$$\mathbf{H} = -\frac{\partial \|\mathbf{X} - \mathbf{\Omega}\mathbf{A}\|_F^2}{\partial \mathbf{\Omega}} = 2\mathbf{X}\mathbf{A}^T - 2\mathbf{\Omega}\mathbf{A}\mathbf{A}^T, \quad (8)$$

the search direction of the  $j$ th row of  $\mathbf{\Omega}$ , i.e. the projection of each row of  $\mathbf{H}$  onto the tangent space of  $\mathcal{S}_{m,1}$ , is [26, pp. 49]

$$\bar{\mathbf{h}}_j = \mathbf{H}_{j,:}(\mathbf{I} - \mathbf{\Omega}_{j,:}^T \mathbf{\Omega}_{j,:}). \quad (9)$$

The  $j$ -th row of  $\mathbf{\Omega}$  is updated along the line search path as follows [26, pp. 103]

$$\mathbf{\Omega}_{j,:}(\alpha) = \begin{cases} \mathbf{\Omega}_{j,:} & \text{if } \|\bar{\mathbf{h}}_j\|_2 = 0, \\ \mathbf{\Omega}_{j,:} \cos(\alpha\|\bar{\mathbf{h}}_j\|_2) + (\bar{\mathbf{h}}_j/\|\bar{\mathbf{h}}_j\|_2) \sin(\alpha\|\bar{\mathbf{h}}_j\|_2) & \\ \text{otherwise,} & \end{cases} \quad (10)$$

where  $\alpha$  is the step size which is determined by the golden section search method [27].

### 3. Proposed Image Restoration Formulation

To simplify the problem, the logarithmic transform is employed here to convert the multiplicative noise model to an additive one, as in [2], [4], [5]. Taking the (element-wise) logarithms of both sides of (1), we have

$$\begin{aligned} \log \mathbf{w} &= \log \mathbf{g} + \log \mathbf{u} \\ \mathbf{z} &= \mathbf{y} + \mathbf{v} \end{aligned} \quad (11)$$

where  $\mathbf{z}$ ,  $\mathbf{y}$  and  $\mathbf{v}$  denote the element-wise logarithms of  $\mathbf{w}$ ,  $\mathbf{g}$ , and  $\mathbf{u}$ , respectively. Since the function  $u = e^v$  is strictly monotonic and the elements of  $\mathbf{u}$  satisfy the i.i.d. Gamma distribution (2), the probability density function of the elements

in  $\mathbf{v}$  is given by [28, pp. 207]

$$\begin{aligned} f_v(v) &= f_u(e^v) \frac{d(e^v)}{dv} \\ &= f_u(e^v) \cdot e^v \\ &= \frac{L^L}{\Gamma(L)} e^{L(v-e^v)}. \end{aligned} \tag{12}$$

Hence, the probability density function of  $\mathbf{v}$  is given by

$$f_{\mathbf{v}}(\mathbf{v}) = \prod_{i=1}^N \frac{L^L}{\Gamma(L)} e^{L(v_i - e^{v_i})}, \tag{13}$$

where  $v_i$  denote the elements of the vector  $\mathbf{v}$  with  $i = 1, 2, \dots, N$ . As a result, the log-likelihood function can be written as

$$\begin{aligned} \log f_{\mathbf{z}|\mathbf{y}}(\mathbf{z}|\mathbf{y}) &= \log f_{\mathbf{v}}(\mathbf{z} - \mathbf{y}) \\ &= N \log \frac{L^L}{\Gamma(L)} + L \sum_{i=1}^N z_i - L \sum_{i=1}^N (y_i + e^{z_i - y_i}). \end{aligned} \tag{14}$$

The *maximum likelihood* (ML) estimate for  $\mathbf{y}$  can be determined by maximizing equation (14) with respect to  $\mathbf{y}$ . Furthermore, by omitting the first two terms which do not depend on  $\mathbf{y}$  and scaling the last term by the negative constant coefficient  $-L$ , the maximization of (14) can be rewritten as the following minimization problem, i.e.

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \sum_{i=1}^N (y_i + e^{z_i - y_i}). \tag{15}$$

It is straightforward to check that the optimal solution to the above problem is  $\hat{\mathbf{y}} = \mathbf{z}$ , but it is an invalid solution for the denoising task. This is due to the over-fitting problem which can be regarded as a general issue of maximum likelihood [29, pp. 9]. In order to avoid this problem, the regularization technique is often employed, which involves adding penalty terms based on the prior information of  $\mathbf{y}$ .

Using the data fidelity term based on the ML estimate (15), the proposed restoration formulation utilizes two regularizations promoting the sparsity and the smoothness prior respectively. The first one is based on the assumption that the image patches are sparse with respect to an analysis dictionary. Since adaptive analysis dictionaries usually have the potential to fit signals better than pre-defined dictionaries [20], the analysis dictionary learned via the Analysis SimCO algorithm is applied in the proposed method. The second regularization term is the smoothness regularizer based on the discrete derivatives of the image, the purpose of which is to smooth the noise further. Combining these two regularizers with the data fidelity term, our new formulation can be written as

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}} \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) + \lambda_1 \|\mathbf{\Omega} \mathbf{Y}\|_1 + \lambda_2 G_\beta\{R(\mathbf{Y})\}, \quad (16)$$

180 where  $\mathbf{\Omega} \in \mathbb{R}^{p \times m}$  denotes the learned analysis dictionary with Algorithm 1. In this formulation, the restored image  $\mathbf{y}$  is expanded as small patches of size  $\sqrt{m} \times \sqrt{m}$  which form the columns of the matrix  $\mathbf{Y} \in \mathbb{R}^{m \times n}$ . As such, the dictionary  $\mathbf{\Omega}$  is learned from image patches instead of the whole image. Similarly, the matrix  $\mathbf{Z} \in \mathbb{R}^{m \times n}$  is obtained from the observed log-image  $\mathbf{z}$ , where  $n$  denotes  
185 the number of image patches.

The data fidelity term  $\sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}})$  is the image patch version of the ML estimate (15). The parameters  $\lambda_1$  and  $\lambda_2$  are the multipliers to balance the data fidelity term and the regularizers. The first regularization term  $\|\mathbf{\Omega} \mathbf{Y}\|_1$ , which is a relaxation of  $\|\mathbf{\Omega} \mathbf{Y}\|_0$ , reflects the sparse property of  
190 the image patches with respect to the dictionary  $\mathbf{\Omega}$ . The second regularization term  $G_\beta\{R(\mathbf{Y})\}$  is used to promote the smoothness of the whole image, where the image patch version  $\mathbf{Y}$  is reshaped back to the complete image by applying the operator  $R(\cdot)$  and  $G_\beta\{\cdot\}$  is the smoothness promotion function.

For a given image denoted by  $\mathbf{S} \in \mathbb{R}^{d \times d}$ ,  $G_\beta\{\mathbf{S}\}$  is defined as

$$G_\beta\{\mathbf{S}\} = \sum_{i=1}^d \sum_{j=1}^d \left( \sqrt{(\nabla_h \mathbf{S}_{i,j})^2 + (\nabla_v \mathbf{S}_{i,j})^2} \right)^\beta, \quad (17)$$

where  $\nabla_h \mathbf{S}_{i,j}$  and  $\nabla_v \mathbf{S}_{i,j}$  denote the horizontal and vertical differences at pixel  $\mathbf{S}_{i,j}$ . More specifically, they are given by the first-order differences between pixel  $\mathbf{S}_{i,j}$  and its horizontal and vertical neighbouring pixels respectively, i.e.

$$\nabla_h \mathbf{S}_{i,j} = \begin{cases} \mathbf{S}_{i+1,j} - \mathbf{S}_{i,j} & \text{if } i < d, \\ 0 & \text{if } i = d. \end{cases} \quad (18)$$

and

$$\nabla_v \mathbf{S}_{i,j} = \begin{cases} \mathbf{S}_{i,j+1} - \mathbf{S}_{i,j} & \text{if } j < d, \\ 0 & \text{if } j = d. \end{cases} \quad (19)$$

The parameter  $\beta$  controls the degree of smoothing. Notice that the smoothness promotion function  $G_\beta\{\cdot\}$  is equivalent to the TV regularizer [14] when  $\beta = 1$ , as such the smoothness regularizer  $G_\beta\{\cdot\}$  can be viewed as a generalization of the TV regularizer.

It should be noted that the proposed image restoration formulation (16) can be regarded as an extension of the model in our previous work [22], which is given by

$$\mathbf{Y}^* = \arg \min_{\mathbf{Y}} \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) + \lambda \|\mathbf{\Omega Y}\|_1. \quad (20)$$

In this model, only the regularizer based on an analysis dictionary is considered. If the multiplier  $\lambda_2$  in the model (16) is set as zero, model (16) will reduce to (20). From this point of view, the restoration model (20) can be seen as a special case of (16). The sparsity based regularizer  $\|\mathbf{\Omega Y}\|_1$  can be regarded as a local prior since it is defined with the image patches in  $\mathbf{Y} \in \mathbb{R}^{m \times n}$  whereas the smoothness regularizer in (16) is a global prior which depends on the entire image  $R(\mathbf{Y}) \in \mathbb{R}^{\sqrt{N} \times \sqrt{N}}$ . Thus, the introduction of the smoothness regularizer not only further reinforces the smoothness of the restored image, but also takes the global prior of the image into consideration.

In general, the proposed method consists of two stages: analysis dictionary learning and image recovery. We refer to this method as Multiplicative

Noise Removal using Analysis Dictionary Learning and a Smoothness Regular-  
 izer (MNR-ADL-SR). In the dictionary learning stage, an analysis dictionary  
 210  $\mathbf{\Omega}$  is learned using the Analysis SimCO algorithm [21] which has already been  
 reviewed in Section 2. The goal of the image recovery stage is to restore the  
 denoised image from the observed image, which is achieved by addressing the  
 optimization problem (16). The restored log-image  $\hat{\mathbf{y}}$  can be obtained by ap-  
 215 plying the operator  $R(\cdot)$  to the solution to (16), and thus the denoised image  
 $\hat{\mathbf{g}}$  can be obtained by taking the exponential transform of  $\hat{\mathbf{y}}$ . The optimization  
 method to address (16) will be presented in the next section.

#### 4. Optimization Method

In this section, we propose a new method to solve the optimization problem  
 220 in (16). Firstly, a variable splitting technique is employed to construct a de-  
 composable structure in the objective function across multiple variables, which  
 results in an equivalent constrained optimization problem. Then the ADMM  
 framework [24] is applied to deal with the constrained optimization problem.

Using the variable splitting technique, the problem (16) can be converted to  
 the equivalent constrained optimization task as follows

$$\begin{aligned} \arg \min_{\{\mathbf{Y}, \mathbf{T}, \mathbf{M}\}} & \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) + \lambda_1 \|\mathbf{T}\|_1 + \lambda_2 G_\beta\{R(\mathbf{M})\} \\ \text{s.t. } & \mathbf{T} = \mathbf{\Omega Y}, \mathbf{M} = \mathbf{Y}. \end{aligned} \quad (21)$$

The variables  $\mathbf{T} = \mathbf{\Omega Y}$  and  $\mathbf{M} = \mathbf{Y}$  are introduced to eliminate  $\mathbf{Y}$  in the  
 225 regularization terms and therefore make the objective function separable with  
 respect to the variables  $\mathbf{Y}, \mathbf{T}, \mathbf{M}$ .

ADMM can be viewed as an attempt to combine the decomposable ben-  
 efit of dual ascent and the superior convergence property of the augmented  
 Lagrangian methods for constrained optimization [24]. The constrained opti-  
 mization problem (21) can be handled with ADMM since the objective function  
 becomes separable across the variables and the decomposed sub-problems are

easier to address. Using dual parameters  $\mathbf{B}_1 \in \mathbb{R}^{p \times n}$  and  $\mathbf{B}_2 \in \mathbb{R}^{m \times n}$ , the augmented Lagrangian can be formed by adding two penalty terms  $\langle \mathbf{B}_1, \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T} \rangle$ ,  $\langle \mathbf{B}_2, \mathbf{Y} - \mathbf{M} \rangle$  and two extra quadratic terms related to the constraints, that is

$$\begin{aligned}
L_{\gamma_1, \gamma_2}(\mathbf{Y}, \mathbf{T}, \mathbf{M}, \mathbf{B}_1, \mathbf{B}_2) &= \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) + \lambda_1 \|\mathbf{T}\|_1 + \lambda_2 G_\beta\{R(\mathbf{M})\} \\
&\quad + \gamma_1 \langle \mathbf{B}_1, \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T} \rangle + \frac{\gamma_1}{2} \|\boldsymbol{\Omega} \mathbf{Y} - \mathbf{T}\|_F^2 + \gamma_2 \langle \mathbf{B}_2, \mathbf{Y} - \mathbf{M} \rangle + \frac{\gamma_2}{2} \|\mathbf{Y} - \mathbf{M}\|_F^2 \\
&= \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) + \lambda_1 \|\mathbf{T}\|_1 + \lambda_2 G_\beta\{R(\mathbf{M})\} \\
&\quad + \frac{\gamma_1}{2} \|\mathbf{B}_1 + \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T}\|_F^2 - \frac{\gamma_1}{2} \|\mathbf{B}_1\|_F^2 + \frac{\gamma_2}{2} \|\mathbf{B}_2 + \mathbf{Y} - \mathbf{M}\|_F^2 - \frac{\gamma_2}{2} \|\mathbf{B}_2\|_F^2,
\end{aligned} \tag{22}$$

where  $\gamma_1, \gamma_2 > 0$  are the penalty coefficients. ADMM alternatively updates each of the variables  $\{\mathbf{Y}, \mathbf{T}, \mathbf{M}, \mathbf{B}_1, \mathbf{B}_2\}$ , while keeping the others fixed. We use the scaled form of ADMM [24] as it is more concise to express. In the  $t$ -th iteration, it consists of the following steps

$$\mathbf{Y}^{(t+1)} = \arg \min_{\mathbf{Y}} L_{\gamma_1, \gamma_2}(\mathbf{Y}, \mathbf{T}^{(t)}, \mathbf{M}^{(t)}, \mathbf{B}_1^{(t)}, \mathbf{B}_2^{(t)}) \tag{23}$$

$$\mathbf{T}^{(t+1)} = \arg \min_{\mathbf{T}} L_{\gamma_1, \gamma_2}(\mathbf{Y}^{(t+1)}, \mathbf{T}, \mathbf{B}_1^{(t)}) \tag{24}$$

$$\mathbf{M}^{(t+1)} = \arg \min_{\mathbf{M}} L_{\gamma_1, \gamma_2}(\mathbf{Y}^{(t+1)}, \mathbf{M}, \mathbf{B}_2^{(t)}) \tag{25}$$

$$\mathbf{B}_1^{(t+1)} = \mathbf{B}_1^{(t)} + (\boldsymbol{\Omega} \mathbf{Y}^{(t+1)} - \mathbf{T}^{(t+1)}) \tag{26}$$

$$\mathbf{B}_2^{(t+1)} = \mathbf{B}_2^{(t)} + (\mathbf{Y}^{(t+1)} - \mathbf{M}^{(t+1)}). \tag{27}$$

In fact, ADMM can be interpreted as reducing the regularized problem (16) to



a sequence of sub-problems which are easier to solve. The ADMM iterations (23)-(27) are performed until the change of  $\mathbf{Y}^{(t+1)}$  is relatively small compared with  $\mathbf{Y}^{(t)}$ . 230

Now we explain the update of variables in (23)-(25) respectively. Ignoring the terms unrelated to  $\mathbf{Y}$ , the minimization problem (23) can be written as

$$\begin{aligned} \arg \min_{\mathbf{Y}} & \sum_{i=1}^m \sum_{j=1}^n (\mathbf{Y}_{i,j} + e^{\mathbf{Z}_{i,j} - \mathbf{Y}_{i,j}}) \\ & + \frac{\gamma_1}{2} \|\mathbf{B}_1 + \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T}\|_F^2 + \frac{\gamma_2}{2} \|\mathbf{B}_2 + \mathbf{Y} - \mathbf{M}\|_F^2. \end{aligned} \quad (28)$$

As this objective function is differentiable, the gradient-based methods can be applied. Here we employ the gradient descent method which has a relatively low computational complexity. The step size can be determined by line search methods [30], however, a small fixed step size also works well, according to our experiments. Given a step size  $\mu$ , the update of  $\mathbf{Y}$  can be written as

$$\mathbf{Y} = \mathbf{Y} - \mu \nabla_{\mathbf{Y}}. \quad (29)$$

The symbol  $\nabla_{\mathbf{Y}}$  denotes the gradient of (28) with respect to  $\mathbf{Y}$ , which can be calculated as follows

$$\nabla_{\mathbf{Y}} = (\mathbf{1} - e^{\mathbf{Z} - \mathbf{Y}}) + \gamma_1 \boldsymbol{\Omega}^T (\mathbf{B}_1 + \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T}) + \gamma_2 (\mathbf{B}_2 + \mathbf{Y} - \mathbf{M}), \quad (30)$$

where  $\mathbf{1} \in \mathbb{R}^{m \times n}$  is an all-one matrix with the same size as  $\mathbf{Y}$  and  $e^{\mathbf{Z} - \mathbf{Y}}$  denotes the element-wise exponential of  $\mathbf{Z} - \mathbf{Y}$ .

For the update of  $\mathbf{T}$ , the problem (24) can be written as

$$\arg \min_{\mathbf{T}} \lambda_1 \|\mathbf{T}\|_1 + \frac{\gamma_1}{2} \|\mathbf{B}_1 + \boldsymbol{\Omega} \mathbf{Y} - \mathbf{T}\|_F^2. \quad (31)$$

Notice that this problem has a closed-form solution given by [24]

$$\mathbf{T} = ST_{\lambda_1/\gamma_1}(\boldsymbol{\Omega} \mathbf{Y} + \mathbf{B}_1). \quad (32)$$

The symbol  $ST_{\lambda_1/\gamma_1}(\cdot)$  represents the element-wise soft-thresholding operator defined by

$$ST_{\lambda_1/\gamma_1}(\theta) = \begin{cases} \theta - \frac{\lambda_1}{\gamma_1} \cdot \text{sgn}(\theta) & \text{if } |\theta| \geq \frac{\lambda_1}{\gamma_1} \\ 0 & \text{otherwise,} \end{cases} \quad (33)$$

where  $\text{sgn}(\theta)$  returns the sign of  $\theta$ .

Dropping the unrelated terms, the update of  $\mathbf{M}$  based on (25) can be obtained by considering the following problem

$$\begin{aligned} \arg \min_{\mathbf{M}} \quad & \lambda_2 G_\beta(\mathbf{M}_R) + \frac{\gamma_2}{2} \|\mathbf{N} - \mathbf{M}\|_F^2 \\ \text{s.t.} \quad & \mathbf{M}_R = R(\mathbf{M}), \end{aligned} \quad (34)$$

where  $\mathbf{N} = \mathbf{B}_2 + \mathbf{Y}$ . In the objective function, there are two variables  $\mathbf{M}_R$  and  $\mathbf{M}$  which are linked via the constraint  $\mathbf{M}_R = R(\mathbf{M})$ . By applying the operator  $R(\cdot)$  to the matrices  $\mathbf{N}$  and  $\mathbf{M}$  respectively, the quadratic term can be rewritten in terms of  $\mathbf{M}_R$ , i.e.

$$\begin{aligned} & \|\mathbf{N} - \mathbf{M}\|_F^2 \\ &= \|(R(\mathbf{N}) - R(\mathbf{M})) \circ \mathbf{C}_m\|_F^2 \\ &= \|(\mathbf{N}_R - \mathbf{M}_R) \circ \mathbf{C}_m\|_F^2, \\ &\approx m \|\mathbf{N}_R - \mathbf{M}_R\|_F^2, \end{aligned} \quad (35)$$

where  $\mathbf{N}_R = R(\mathbf{N})$ , and  $\mathbf{C}_m$  is a constant matrix depending on the operator  $R(\cdot)$  and  $m$ . Specifically, the squares of the elements in  $\mathbf{C}_m$  represent the number of times that the corresponding elements of  $\mathbf{N}_R - \mathbf{M}_R$  appear in the matrix  $\mathbf{N} - \mathbf{M}$ . When the overlap between two neighbouring patches is  $\sqrt{m} - 1$ , most elements in  $\mathbf{C}_m$ , except for the elements on the border, take the same value  $\sqrt{m}$ . Hence, the quadratic term  $\|\mathbf{N} - \mathbf{M}\|_F^2$  can be approximated as  $m \|\mathbf{N}_R - \mathbf{M}_R\|_F^2$ .

The details of the approximation (35) are illustrated in Fig. 1, using a specific example. As shown in the figure, the inverse operator of  $R(\cdot)$  transforms the matrix  $\mathbf{D}_R$  to its patch version by extracting overlapping patches of size

$\sqrt{m} \times \sqrt{m}$  and reshaping each patch as one column, with  $m = 4$ . Note that  
 “Col” in Fig. 1 is short for “Columns”. Due to the overlap among the patches,  
 the elements of  $\mathbf{D}_R$  appear in  $\mathbf{D}$  for various times, and the number of times that  
 these elements appear can be represented by the squares of the corresponding  
 elements of the constant matrix  $\mathbf{C}_4$  which is presented in the right part of Fig.  
 1. Therefore, we have  $\|\mathbf{D}\|_F^2 = \sum_{i,j} c_{ij} d_{ij}^2 = \|\mathbf{D}_R \circ \mathbf{C}_4\|_F^2$ , where  $c_{ij}$  denotes  
 the appearance times of the element  $d_{ij}$  in  $\mathbf{D}$ . As most elements of  $\mathbf{C}_4$  are  
 $\sqrt{m} = 2$ ,  $\|\mathbf{D}_R \circ \mathbf{C}_4\|_F^2$  can be approximated as  $m\|\mathbf{D}_R\|_F^2$ . Generalizing this  
 specific example, the approximation equation (35) can be obtained.

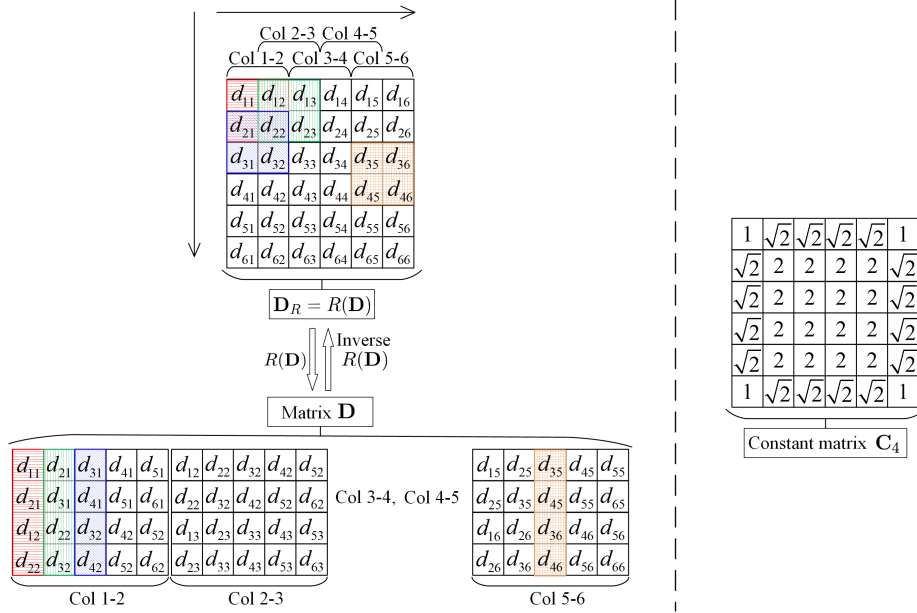


Figure 1: This figure presents a specific example to illustrate the approximation equation (35). “Col” is short for “columns”. In this figure,  $\mathbf{D}_R$  denotes the result of applying the operator  $R(\cdot)$  to the matrix  $\mathbf{D}$ . Four patches extracted from  $\mathbf{D}_R$  are filled using different colors and the corresponding columns in  $\mathbf{D}$  are highlighted in the same colors, respectively.

As a result, the problem (34) can be relaxed as

$$\mathbf{M}_R^* = \arg \min_{\mathbf{M}_R} \lambda_2 G_\beta(\mathbf{M}_R) + \frac{\gamma_2}{2} \|\mathbf{N}_R - \mathbf{M}_R\|_F^2, \quad (36)$$

and  $\mathbf{M}$  can be obtained by applying the inverse operator of  $R(\cdot)$  to  $\mathbf{M}_R$ .

Obviously, the optimization of (36) depends on the value of  $\beta$ . Here, two cases are considered, i.e.  $\beta \in \{1, 2\}$ . When  $\beta = 1$ , (36) can be written as

$$\arg \min_{\mathbf{M}_R} \lambda_2 \|\mathbf{M}_R\|_{TV} + \frac{\gamma_2}{2} \|\mathbf{N}_R - \mathbf{M}_R\|_F^2. \quad (37)$$

This can be viewed as a TV- $\ell_2$  minimization problem which can be addressed by Chambolle's algorithm [31].

When  $\beta = 2$ , the problem (36) is equivalent to

$$\arg \min_{\mathbf{M}_R} \lambda_2 \sum_{i,j} [(\nabla_h(\mathbf{M}_R)_{i,j})^2 + (\nabla_v(\mathbf{M}_R)_{i,j})^2] + \frac{\gamma_2}{2} \|\mathbf{N}_R - \mathbf{M}_R\|_F^2 \quad (38)$$

and it can be addressed by solving the Euler-Lagrange equation numerically [14], [32]. Specifically, the optimal solution can be approached iteratively by the gradient descent step [32] as follows (the detailed derivation is given in the Appendix)

$$\mathbf{M}_R^{(k+1)} = \mathbf{M}_R^{(k)} + \tau \left[ 2\lambda_2 \left( \frac{\partial}{\partial x} \left( \frac{\partial \mathbf{M}_R^{(k)}}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{\partial \mathbf{M}_R^{(k)}}{\partial y} \right) \right) - \gamma_2 (\mathbf{M}_R^{(k)} - \mathbf{N}_R) \right], \quad (39)$$

where  $\frac{\partial}{\partial x} \left( \frac{\partial \mathbf{M}_R}{\partial x} \right)$  and  $\frac{\partial}{\partial y} \left( \frac{\partial \mathbf{M}_R}{\partial y} \right)$  denote the second-order discrete derivatives in the horizontal and vertical directions respectively,  $\tau$  is the step size and  $k$  represents the iteration number.

To sum up, the optimization for the image reconstruction model (16) can be summarized as Algorithm 2.

#### 4.1. Computational Complexity

The proposed reconstruction method involves the update of the variables  $\mathbf{Y}$ ,  $\mathbf{T}$ ,  $\mathbf{M}$ ,  $\mathbf{B}_1$  and  $\mathbf{B}_2$  in each iteration, as summarized in Algorithm 2. To analyze the time complexity of the proposed method, multiplication is considered as the basic operation. As the update of  $\mathbf{M}$  and  $\mathbf{B}_2$  does not involve multiplications, the time complexity of these two steps can be omitted. In the step for updating  $\mathbf{Y}$ , the computation of the gradient  $\nabla_{\mathbf{Y}}$  is the dominant part, and it requires

---

**Algorithm 2** Image Reconstruction

---

**Input:**  $\mathbf{Z}, \mathbf{\Omega}, \lambda_1, \lambda_2, \beta$

**Output:**  $\mathbf{Y}^*$

**Initialization:**

Initialize the iteration counter  $t = 1$  and the initial point  $\mathbf{Y}^{(t)} = \mathbf{Z}$ .  
Perform the following steps.

**Main Iterations:**

1. Update  $\mathbf{Y}^{(t+1)} \leftarrow \mathbf{Y}^{(t)}$  by applying the gradient descent method to the problem (28), i.e. using equations (29) and (30).
  2. Update  $\mathbf{T}^{(t+1)} \leftarrow \mathbf{T}^{(t)}$  by solving the problem (31) via soft-thresholding (32), (33).
  3. Update  $\mathbf{M}^{(t+1)} \leftarrow \mathbf{M}^{(t)}$  by addressing the problem (36) and applying the inverse operator of  $R(\cdot)$  to the optimal solution  $\mathbf{M}_R^*$ . When  $\beta = 1$ ,  $\mathbf{M}_R^*$  can be obtained by applying Chambolle's algorithm to (37). When  $\beta = 2$ ,  $\mathbf{M}_R^*$  is estimated with the iteration (39).
  4. Update  $\mathbf{B}_1^{(t+1)} \leftarrow \mathbf{B}_1^{(t)}$  and  $\mathbf{B}_2^{(t+1)} \leftarrow \mathbf{B}_2^{(t)}$  based on equations (26) and (27).
  5. If the stopping criterion is satisfied,  $\mathbf{Y}^* = \mathbf{Y}^{(t+1)}$ , quit the iteration. Otherwise, increase the iteration counter  $t = t + 1$  and go back to step 1).
- 

$O(pmn)$  operations with pre-computed  $\mathbf{\Omega}^T \mathbf{\Omega}$  and  $p > m$ . The complexity of the steps for updating  $\mathbf{T}$  and  $\mathbf{B}_2$  is dominated by the computation of  $\mathbf{\Omega} \mathbf{Y}$  which requires  $O(pmn)$  operations. As a result, the total time complexity of each iteration of the proposed method scales as  $O(pmn)$ .

270

## 5. Experimental Results

In this section, the experiments for synthetic images with multiplicative noise and real SAR images are presented respectively. The proposed MNR-ADL-SR algorithm is tested with  $\beta = 1$  and  $\beta = 2$ , which are referred to as MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> respectively <sup>7</sup>. Actually, these two cases can be regarded as two different denoising models and they have different effects in smoothing images. In particular, as has been mentioned in Section 4, when

275

---

<sup>7</sup>The codes of the proposed methods are available from <https://github.com/jd0710/MNR-ADL-SR>

$\beta = 1$ , the smoothness regularizer is equivalent to the TV regularizer [31], [14] which preserves edges while removing noise. In contrast, for the case  $\beta = 1$ , the smoothness regularizer reduces to the isotropic diffusion model [33] which has been shown to be effective in restraining noise, but leads to blurred edges. The reason that  $\beta = 1$  and  $\beta = 2$  are tested in the experiments is to investigate the effect of these two models when they are embedded in the reconstruction model. These two cases are compared with our previous work RSN-ADL [22] and three other recent algorithms: DFN [2] (which outperforms the AA [11] and SO [12] algorithms), MIDAL [4], and MNR-DL-TV-2 [5]<sup>8</sup>. These three algorithms were selected as baselines because of the involvement of sparsity or TV regularizer in their formulations and the availability of their code.



Figure 2: Training images used to learn analysis dictionaries.

For the proposed MNR-ADL-SR<sub>1</sub>, MNR-ADL-SR<sub>2</sub> and RSN-ADL [22] algorithms, the images in Fig. 2 were used as the training data to learn analysis dictionaries. Specifically, the training samples employed to learn the analysis dictionary  $\mathbf{\Omega}$  were the logarithmic transforms of 20000 patches that were extracted randomly from these training images. The size of the training patches was  $8 \times 8$ . The dictionary was initialized as the finite difference operator [18], [20]. The dictionary size is 128 and the co-sparsity for dictionary learning was set as  $l = 100$ . The Analysis SimCO algorithm was performed with 2000 iterations. These parameters were set empirically to be consistent with the work in [21].

<sup>8</sup>The codes for the DFN and MIDAL algorithms were downloaded from <https://fadili.users.greyc.fr/software.html> and <http://www.lx.it.pt/~bioucas/publications.html> respectively. We thank the authors of [5] for sharing their code via email.

## 5.1. Experiments with Synthetic Images

### 5.1.1. Experiment Settings

Four test images: “Cameraman”, “Nîmes”, “Fields” and “Peppers” were employed, which are illustrated in Fig. 3. These images are commonly used to evaluate the algorithms for removing multiplicative noise [2], [4], [5]. The size of the Cameraman and Peppers images is  $256 \times 256$  and the size of Nîmes and  
 305 Fields is  $512 \times 512$ . The grey-scales of all the test images are normalized so that they are in the range  $[1 \ 256]$ . The synthetic noisy images were generated by multiplying the pixels of the original images by i.i.d. Gamma random variables (cf. equations (1) and (2)), with different parameters  $L \in \{10, 4, 1\}$ . The synthetic noisy images are shown in Fig. 4, with the noise level increasing from top to bottom.



Figure 3: Test images: Cameraman, Nîmes, Fields and Peppers.

310

### 5.1.2. Performance Metrics

The denoising performance is evaluated with three quantities: Peak Signal-to-Noise Ratio (PSNR), Mean Absolute-deviation Error (MAE), and the Mean Structural SIMilarity index (MSSIM) [34]. The PSNR and MAE indices have been widely used for the quality assessment of multiplicative noise removal algorithms [2], [4], [5], due to their simplicity and clear physical meanings. For a clean image  $\mathbf{g} \in \mathbb{R}^N$ , the PSNR of its denoised version  $\hat{\mathbf{g}} \in \mathbb{R}^N$  is defined as

$$\text{PSNR} = 10 \log_{10} \frac{N |\max(\mathbf{g}) - \min(\mathbf{g})|^2}{\|\hat{\mathbf{g}} - \mathbf{g}\|_2^2} \quad (\text{in dB}) \quad (40)$$

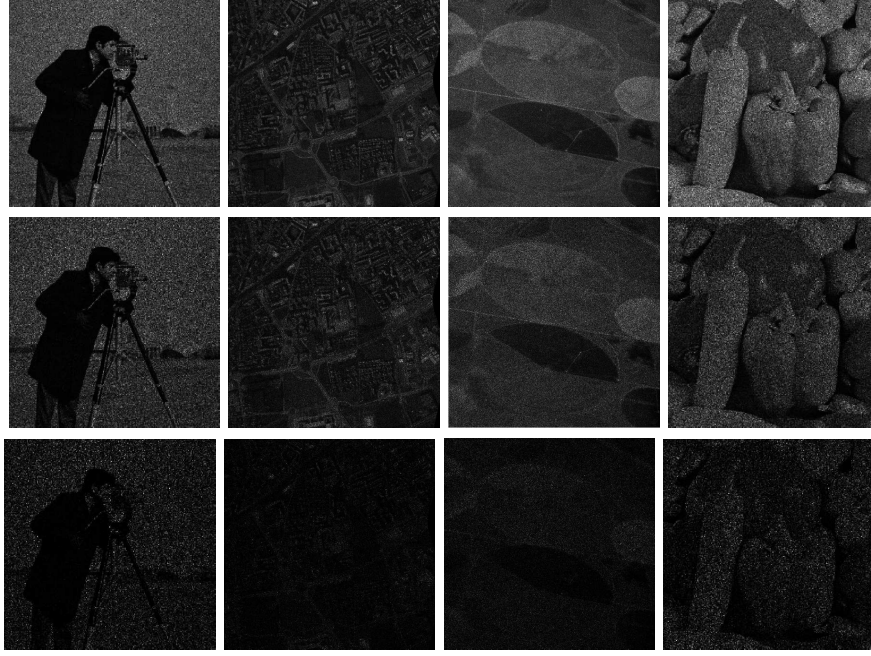


Figure 4: Synthetic noisy images. Top row:  $L = 10$ . Middle row:  $L = 4$ . Bottom row:  $L = 1$ .

where  $\max(\cdot)$  and  $\min(\cdot)$  return the maximum value and the minimum value contained within their operands respectively. The MAE is given by

$$\text{MAE} = \frac{1}{N} \|\hat{\mathbf{g}} - \mathbf{g}\|_1. \quad (41)$$

As indicated by the definitions above, both PSNR and MAE can be regarded as the error-based measurements which are determined by the pixel-to-pixel differences between the denoised image and the reference image. They are useful to obtain general performance assessments on the whole image, but they consider little information about the preservation of specific features so that their evaluations are not very consistent with the perceptual quality. In contrast, the MSSIM index stresses the preservation quality of structural information and is able to reflect the visual perception of humans better [34]. The value of MSSIM ranges over the interval  $[0, 1]$ , with 1 indicating perfect structure similarity. The same set of parameters as originally suggested in [34] is employed.



### 5.1.3. The Selection of the Regularization Parameters

For the proposed algorithm, the selection of the regularization parameters  $\lambda_1$  and  $\lambda_2$  is critical. These two parameters are the coefficients of the learned analysis dictionary based regularizer and the smoothness regularizer, respectively. They are used to balance the data fidelity term and the regularization terms, and thus the appropriate settings of these two parameters depend on the relative importance of the three terms in the reconstruction model. In particular, the importance of the data fidelity term depends on the level of the noise, and the importance of the regularizers depends on the characteristics of the images. Taking the Cameraman image as an example, the PSNR results obtained by MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> using different regularization parameters are demonstrated in Fig. 5. The subfigures from top to bottom display the results with the noise levels  $L = 10, 4, 1$  respectively. The left column shows the results of MNR-ADL-SR<sub>1</sub> and the right column gives the PSNR values obtained by MNR-ADL-SR<sub>2</sub>. Fig. 5 shows that the performance of the proposed algorithms varies with the regularization coefficients  $\lambda_1$  and  $\lambda_2$ . The changing patterns of MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> are similar to each other in general. In the cases of  $L = 10$  and  $L = 4$ , when  $\lambda_1$  is set as a relatively small value, the increase of  $\lambda_2$  leads to an improvement in PSNR to some point followed by a reduction in PSNR. When the value of  $\lambda_1$  is relatively large, the PSNR will decrease with the increase of  $\lambda_2$  and the rate of the decrease of MNR-ADL-SR<sub>2</sub> is slower than that of MNR-ADL-SR<sub>1</sub>. In the  $L = 1$  case, a relatively large  $\lambda_2$  does not result in such a degraded PSNR as when  $L = 10$  and  $L = 4$ , especially for MNR-ADL-SR<sub>2</sub>.

Based on our experiments, some general guidelines could be given to the settings of  $\lambda_1$  and  $\lambda_2$ . The setting of  $\lambda_1$  mainly depends on the level of the noise. For a higher noise level,  $\lambda_1$  should be set as a larger value. Based on our experimental tests, the suggested intervals from which  $\lambda_1$  can be chosen as a function of noise level are summarized Table 3. Appropriate setting of  $\lambda_2$  mainly depends on the amount of texture areas and smooth areas in the image.

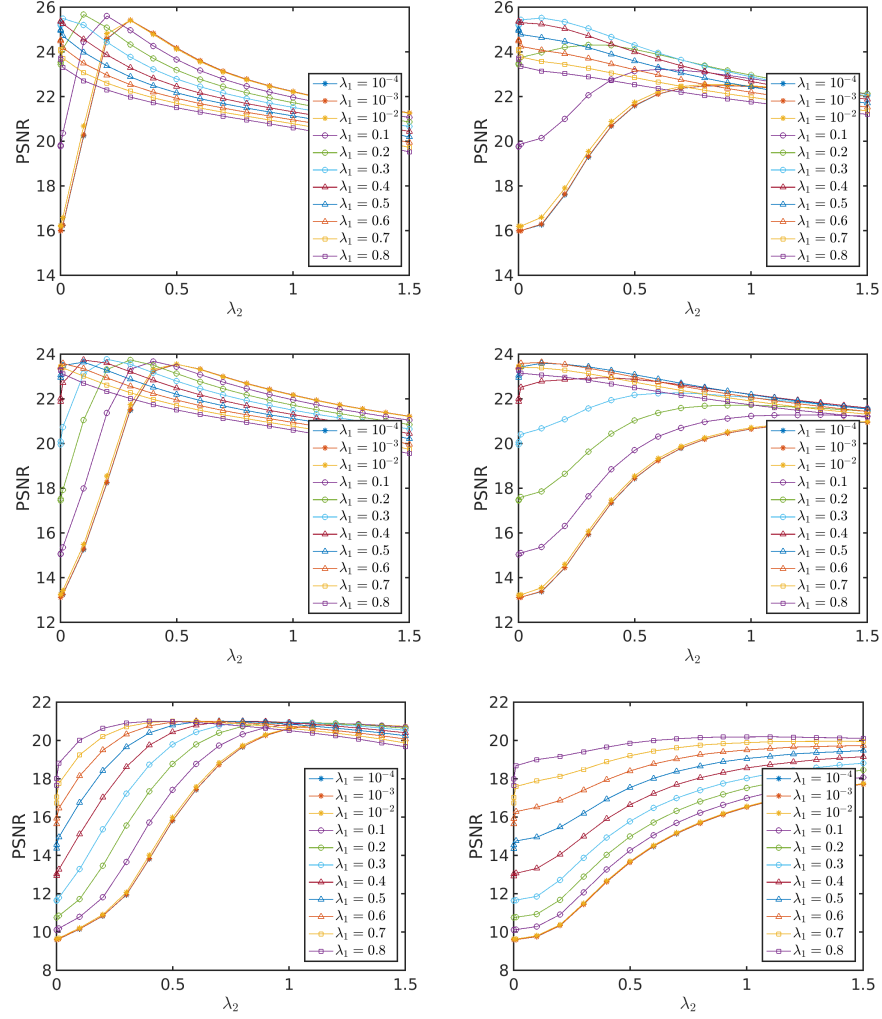


Figure 5: PSNR results for the Cameraman image with different regularization parameters. Left column: MNR-ADL-SR<sub>1</sub>. Right column: MNR-ADL-SR<sub>2</sub>. Top row:  $L = 10$ . Middle row:  $L = 4$ . Bottom row:  $L = 1$ .

Table 1: Parameters used in the algorithms for “Cameraman” and “Nimes”

L	Algorithm	Cameraman	Nimes
10	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 0.2, \lambda_2 = 0.1$	$\lambda_1 = 0.2, \lambda_2 = 0.01$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 0.3, \lambda_2 = 0.1$	$\lambda_1 = 0.2, \lambda_2 = 0.2$
	RSN-ADL	$\lambda = 0.4$	$\lambda = 0.3$
	MIDAL	$\lambda = 6.1$	$\lambda = 4$
	MNR-DL-TV-2	$\lambda = 2.7$	$\lambda = 17.5$
	DFN	$T = 2.1\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.3, \lambda_1 = 10$	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.3, \lambda_1 = 10$
4	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 0.3, \lambda_2 = 0.2$	$\lambda_1 = 0.4, \lambda_2 = 0.001$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 0.6, \lambda_2 = 0.1$	$\lambda_1 = 0.3, \lambda_2 = 0.4$
	RSN-ADL	$\lambda = 0.7$	$\lambda = 0.5$
	MIDAL	$\lambda = 4.5$	$\lambda = 2.7$
	MNR-DL-TV-2	$\lambda = 1.2$	$\lambda = 13.5$
	DFN	$T = 2.5\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.8, \lambda_1 = 5.7$	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.5, \lambda_1 = 10$
1	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 0.5, \lambda_2 = 0.7$	$\lambda_1 = 1.2, \lambda_2 = 10^{-4}$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 1.3, \lambda_2 = 0.2$	$\lambda_1 = 1.2, \lambda_2 = 10^{-4}$
	RSN-ADL	$\lambda = 1.6$	$\lambda = 1.2$
	MIDAL	$\lambda = 2.7$	$\lambda = 2$
	MNR-DL-TV-2	$\lambda = 0.01$	$\lambda = 3.2$
	DFN	$T = 2.6\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.8, \lambda_1 = 5.7$	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.5, \lambda_1 = 10$

Table 2: Parameters used in the algorithms for “Fields” and “Peppers”

L	Algorithm	Fields	Peppers
10	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 0.4, \lambda_2 = 0.01$	$\lambda_1 = 0.1, \lambda_2 = 0.2$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 0.4, \lambda_2 = 0.01$	$\lambda_1 = 0.4, \lambda_2 = 0.01$
	RSN-ADL	$\lambda = 0.5$	$\lambda = 0.4$
	MIDAL	$\lambda = 6.7$	$\lambda = 5.9$
	MNR-DL-TV-2	$\lambda = 3.7$	$\lambda = 4.2$
	DFN	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.3, \lambda_1 = 10$	$T = 1.8\sqrt{\Psi_1(L)},$ $\lambda_0 = 0.9, \lambda_1 = 5$
4	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 0.7, \lambda_2 = 0.01$	$\lambda_1 = 0.2, \lambda_2 = 0.3$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 0.7, \lambda_2 = 0.01$	$\lambda_1 = 0.6, \lambda_2 = 0.01$
	RSN-ADL	$\lambda = 1$	$\lambda = 0.7$
	MIDAL	$\lambda = 4.5$	$\lambda = 4.1$
	MNR-DL-TV-2	$\lambda = 0.7$	$\lambda = 2.2$
	DFN	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.3, \lambda_1 = 10$	$T = 1.9\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.1, \lambda_1 = 3.5$
1	MNR-ADL-SR <sub>1</sub>	$\lambda_1 = 1.8, \lambda_2 = 0.01$	$\lambda_1 = 1.6, \lambda_2 = 0.001$
	MNR-ADL-SR <sub>2</sub>	$\lambda_1 = 1.8, \lambda_2 = 0.01$	$\lambda_1 = 1.6, \lambda_2 = 0.001$
	RSN-ADL	$\lambda = 3.5$	$\lambda = 1.6$
	MIDAL	$\lambda = 3.5$	$\lambda = 2.4$
	MNR-DL-TV-2	$\lambda = 0.01$	$\lambda = 0.01$
	DFN	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1.2, \lambda_1 = 10$	$T = 2\sqrt{\Psi_1(L)},$ $\lambda_0 = 1, \lambda_1 = 3.5$

Table 3: General guidelines for the setting of  $\lambda_1$ 

L	Suggested Interval for $\lambda_1$
10	[0.1, 0.4]
4	[0.2, 0.7]
1	[0.5, 1.8]

Table 4: Performance decrease in PSNR using the tuned parameters for Cameraman

L	Algorithm	Nîmes	Fields	Peppers
10	MNR-ADL-SR <sub>1</sub>	0.41	0.15	0.03
	MNR-ADL-SR <sub>2</sub>	0.53	0.51	0.69
4	MNR-ADL-SR <sub>1</sub>	0.47	0.26	0.04
	MNR-ADL-SR <sub>2</sub>	0.73	0.17	0.45
1	MNR-ADL-SR <sub>1</sub>	0.84	0.33	0.51
	MNR-ADL-SR <sub>2</sub>	0.80	0.41	0.97

In general,  $\lambda_2$  can be chosen from the interval  $[10^{-4}, 0.7]$ . For images containing more texture areas, such as the test image Nîmes, a small  $\lambda_2$  is preferred. For images containing more smooth areas, such as the test image Cameraman, a larger  $\lambda_2$  will give better results.

In our experiments, the regularization coefficients for MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> algorithms were selected empirically. Likewise, the parameters of RSN-ADL [22] and MNR-DL-TV-2 [5] were also determined in this way. The parameters of the MIDAL [4] and DFN [2] algorithms for the first three test images were set as in their original papers and for the Peppers image the parameters are manually tuned to lead to the best PSNR. The parameters of the algorithms used in our experiments are summarized in Tables 1 and 2.

To investigate the stability of the performance of the proposed methods with respect to the choice of the parameters, the parameters tuned for the Cameraman image (as shown in Table 1 and Table 2) were also employed for other test images. The denoising results are compared with those obtained with the tuned parameters for each individual image, and the decreases in PSNR are summarized in Table 4. This table shows that there will be some compromise in performance if the parameters were not tuned to the specific images.

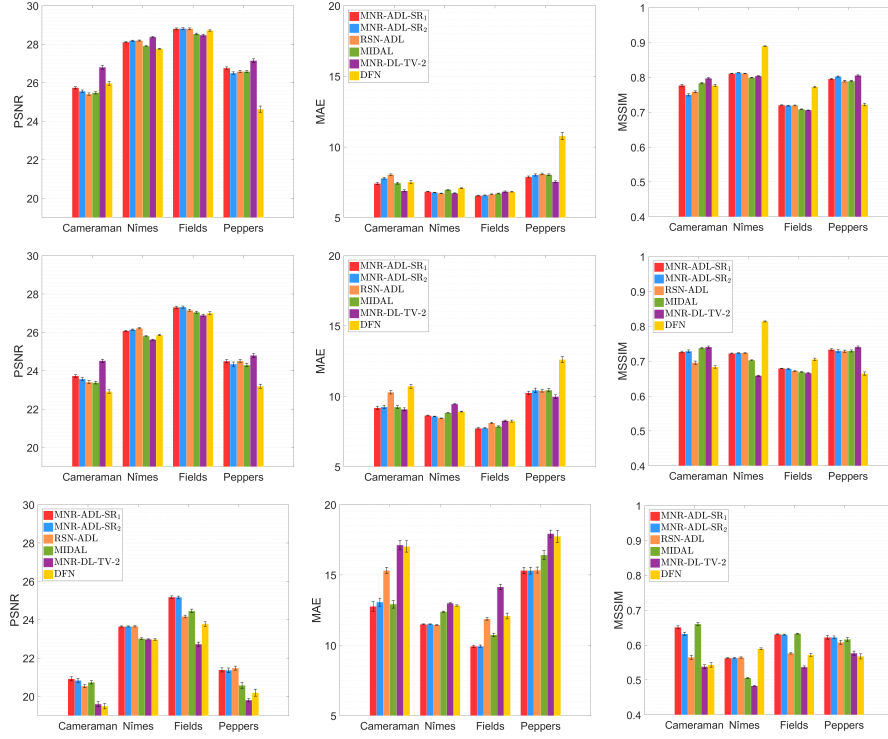


Figure 6: Denoising results in PSNR, MAE and MSSIM based on 30 noisy realizations for each case. Top row:  $L = 10$ . Middle row:  $L = 4$ . Bottom row:  $L = 1$ . (Note that legends are identical for all plots, but omitted in two figures to retain clarity.)

#### 5.1.4. Experimental Results

The algorithms were tested with 30 noise realizations for each case. The samples of the denoising results found in one test are shown in Figs. 7-12. The average results over the 30 random noise realizations measured in PSNR, MAE and MSSIM are provided in Fig. 6, where the bars illustrate the mean results and the error bars display the standard deviations. From the top row to the bottom row, the noise levels are  $L = 10, 4, 1$  respectively.

Let us just compare the MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> algorithms with our previous work RSN-ADL. Generally, it can be seen from the denoised images that the visual appearance of the results obtained by MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> is better than that of RSN-ADL. MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> can preserve image details as well as RSN-ADL, but reconstruc-

t smooth areas better than RSN-ADL (see the background of the Cameraman image in Figs. 7-9 as an example). This demonstrates the benefit of the introduction of the smoothness regularizer. For the Nîmes image, the proposed algorithms do not show visible advantages as compared with RSN-ADL. This could be caused by the fact that Nîmes contains many structural details for which the smoothness regularizers contribute little. The results of MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> are very similar in visual quality and the former algorithm can obtain slightly better results in terms of the performance metrics, as shown in Fig. 6.

For the images Cameraman and Peppers with noise levels  $L = 10$  and  $L = 4$ , the results of MNR-DL-TV-2 are the best in terms of the metrics. However, the proposed algorithms can better preserve fine textures (see the building details in the background of Cameraman in Figs. 7 and 8). For Nîmes and Fields, the proposed algorithms outperform the baseline algorithms for most cases in terms of PSNR and MAE. The DFN algorithm obtains the best MSSIM values for these two images, but some artifacts are also introduced as can be seen in Fig. 11. The denoised images obtained by the MIDAL algorithm have the stair-casing effect, especially when the noise level is high (see Fig. 9). As can be seen in Fig. 6, when  $L = 1$ , the proposed algorithms obtain the best results for most cases, which indicates their superiority in removing a relatively high level of multiplicative noise, as compared with the baseline algorithms.

##### 5.1.5. Comparison with Additive Noise Removal Methods

Since the multiplicative noise is converted to additive noise by applying the logarithmic transform in the proposed methods (i.e. equation (11)), the proposed methods are also compared with classical denoising methods dealing with additive noise. In particular, the TV [31] and K-SVD [17] denoising methods are employed as baselines. In our experiments, the images were denoised using these additive noise removal methods in the log-domain, and then the exponential transform was applied to obtain the denoised images. The parameters of the additive noise removal methods were selected empirically for obtaining

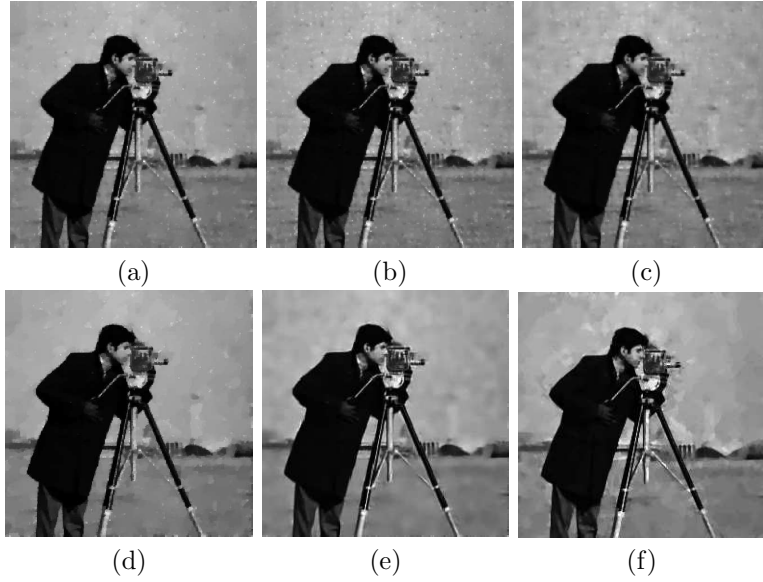


Figure 7: Results for Cameraman ( $L = 10$ ). (a) MNR-ADL-SR<sub>1</sub> (25.67 dB). (b) MNR-ADL-SR<sub>2</sub> (25.52 dB). (c) RSN-ADL (25.36 dB). (d) MIDAL (25.40 dB). (e) MNR-DL-TV-2 (26.62 dB). (f) DFN (26.04 dB).

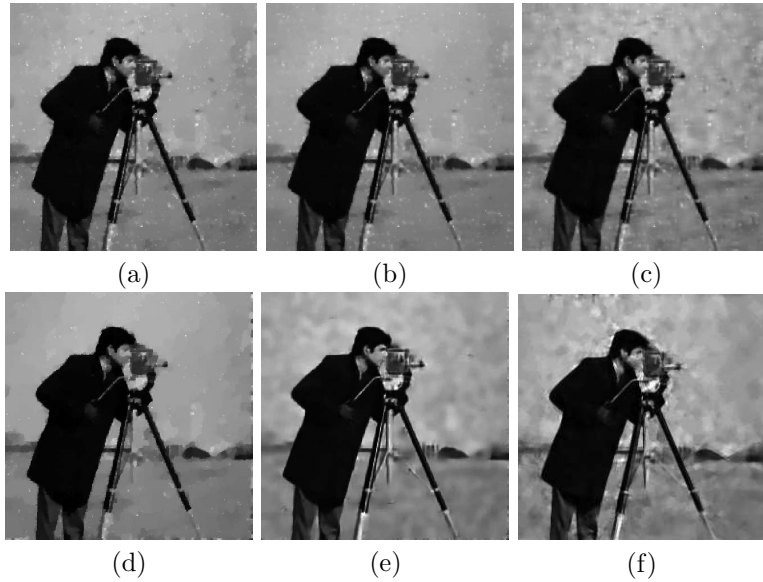


Figure 8: Results for Cameraman ( $L = 4$ ). (a) MNR-ADL-SR<sub>1</sub> (23.65 dB). (b) MNR-ADL-SR<sub>2</sub> (23.51 dB). (c) RSN-ADL (23.35 dB). (d) MIDAL (23.26 dB). (e) MNR-DL-TV-2 (24.52 dB). (f) DFN (23.02 dB).

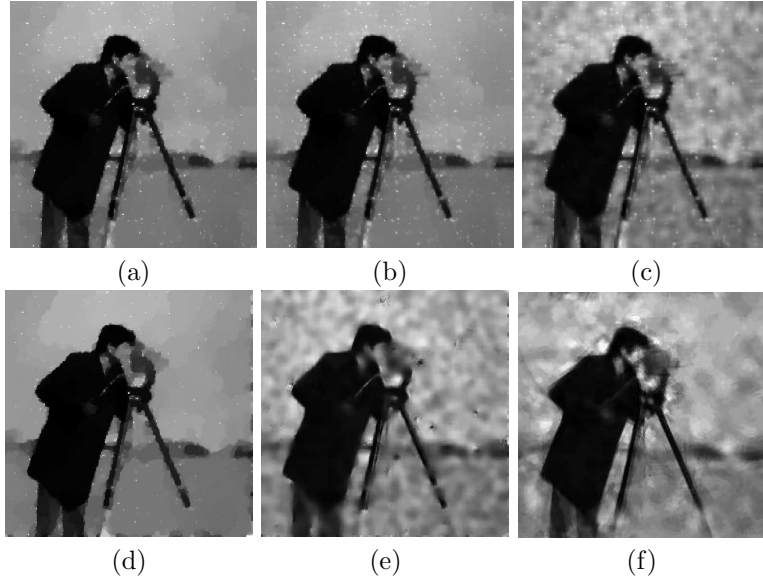


Figure 9: Results for Cameraman ( $L = 1$ ). (a) MNR-ADL-SR<sub>1</sub> (20.97 dB). (b) MNR-ADL-SR<sub>2</sub> (20.89 dB). (c) RSN-ADL (20.62 dB). (d) MIDAL (20.86 dB). (e) MNR-DL-TV-2 (19.69 dB). (f) DFN (19.44 dB).

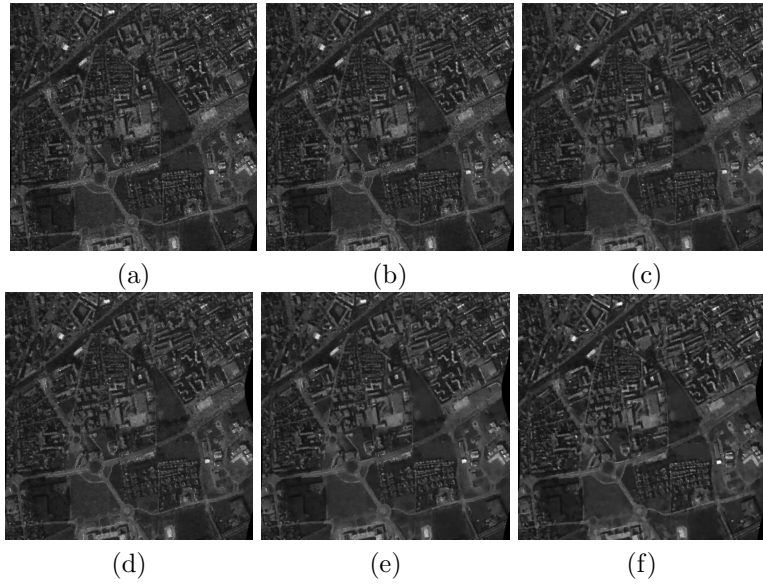


Figure 10: Results for Nîmes ( $L = 10$ ). (a) MNR-ADL-SR<sub>1</sub> (28.14 dB). (b) MNR-ADL-SR<sub>2</sub> (28.21 dB). (c) RSN-ADL (28.22 dB). (d) MIDAL (27.93 dB). (e) MNR-DL-TV-2 (28.42 dB). (f) DFN (27.73 dB).



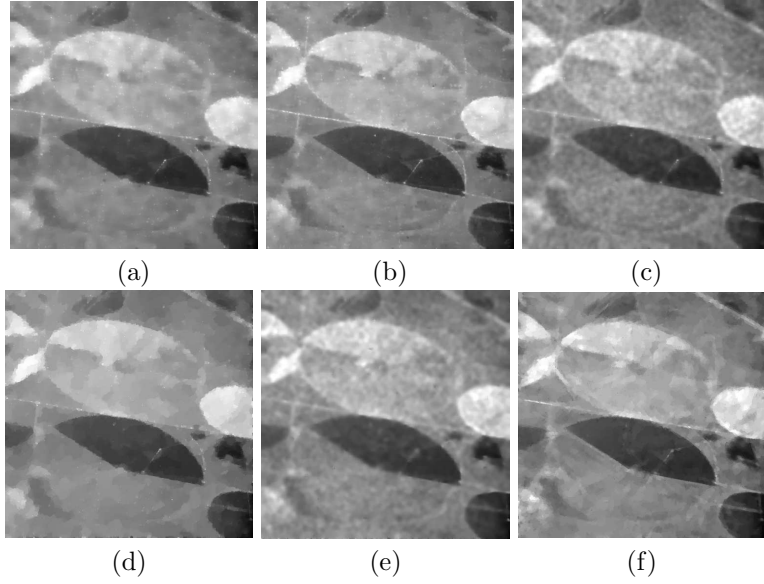


Figure 11: Results for Fields ( $L = 4$ ). (a) MNR-ADL-SR<sub>1</sub> (27.33 dB). (b) MNR-ADL-SR<sub>2</sub> (27.31 dB). (c) RSN-ADL (27.15 dB). (d) MIDAL (27.06 dB). (e) MNR-DL-TV-2 (26.81 dB). (f) DFN (26.93 dB).

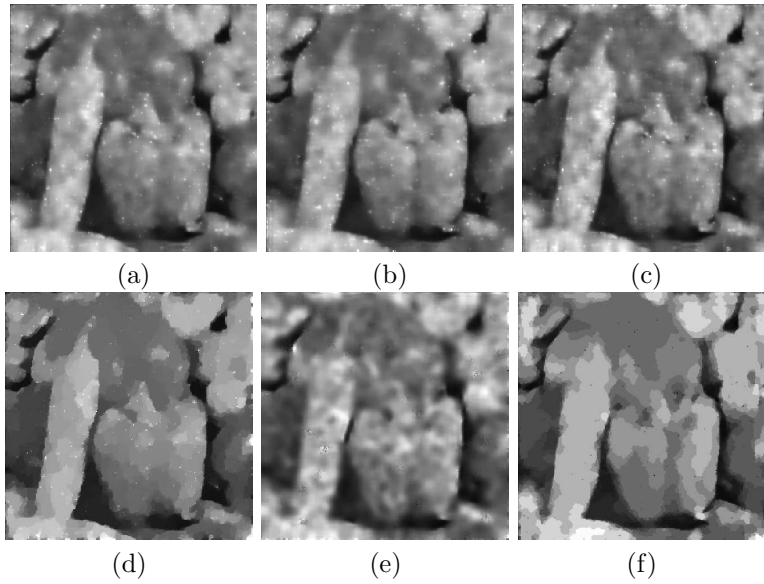


Figure 12: Results for Peppers ( $L = 1$ ). (a) MNR-ADL-SR<sub>1</sub> (21.27 dB). (b) MNR-ADL-SR<sub>2</sub> (21.17 dB). (c) RSN-ADL (21.36 dB). (d) MIDAL (20.53 dB). (e) MNR-DL-TV-2 (19.65 dB). (f) DFN (20.10 dB).

Table 5: PSNR decrease of denoised images obtained using additive noise removal methods.

L	Algorithm	Cameraman	Nimes	Fields	Peppers
10	TV	1.86	0.65	2.52	1.73
	K-SVD	4.46	2.26	3.08	5.19
4	TV	3.79	1.46	5.92	3.79
	K-SVD	5.52	2.72	5.88	5.42
1	TV	8.83	4.80	13.28	8.99
	K-SVD	8.81	5.42	13.35	8.88

highest PSNRs. It has been found that these methods do not outperform the proposed methods, and the decreases in PSNR of these methods as compared with the proposed methods are summarized in Table 5.

## 5.2. Experiments with Real SAR Images

In this subsection, the algorithms are applied to removing speckle noise in the real SAR images<sup>9</sup> shown in Fig. 13. Due to the lack of reference clean images, the metrics used in the experiments for synthetic data can no longer be employed to assess the despeckling performance. For the homogeneous areas where the scene variation is supposed to be negligible, as in the regions marked with red rectangles in Fig. 13, the Equivalent Number of Looks (ENL) measure is suitable for evaluating the level of smoothing [35]. For a given homogeneous region  $\hat{\mathbf{g}}_{reg}$ , the ENL can be computed as

$$\text{ENL} = \frac{[\text{E}(\hat{\mathbf{g}}_{reg})]^2}{\text{Var}(\hat{\mathbf{g}}_{reg})}, \quad (42)$$

where  $\text{E}(\hat{\mathbf{g}}_{reg})$  and  $\text{Var}(\hat{\mathbf{g}}_{reg})$  denote the mean and the variation of the pixel values in region  $\hat{\mathbf{g}}_{reg}$ . This quantity increases as the level of smoothing improves.

For a relatively fair comparison, the same parameters as used for Cameraman with the noise level  $L = 4$  (see Table 1) were employed for the real SAR images.

<sup>9</sup>The test SAR images were downloaded from <https://github.com/zhangyiwei79/Opticks-SAR/tree/master/SAR%20images>



Figure 13: Original SAR images.

The denoised images are shown in Figs. 14-17. The ENL values of the four regions as marked in Fig. 13 are summarized in Table 6.

Table 6: ENL for the homogeneous regions in the denoised SAR images

Algorithm	Region 1	Region 2	Region 3	Region 4
Original	22	29	8	4
MNR-ADL-SR <sub>1</sub>	1538	124100	31460	62
MNR-ADL-SR <sub>2</sub>	1376	43902	15205	64
RSN-ADL	779	4483	3260	47
MIDAL	2345	15516	115800	215
MNR-DL-TV-2	688	1077	71	8
DFN	144	106	36	12

From Figs. 14-17, we can see that all the algorithms are capable of reducing the speckle noise in the SAR images. However, there is still some visible speckle noise in the denoised versions obtained via MNR-DL-TV-2 and DFN (see sub-figures (e) and (f) of Figs. 14-17). The homogeneous areas in the results for the MIDAL algorithm are well-smoothed, which is also confirmed by the large ENL values in Table 6, but some texture details are over-smoothed as shown in the subfigures (d) of Figs. 14 and 17. The proposed MNR-ADL-SR<sub>1</sub> and MNR-ADL-SR<sub>2</sub> algorithms have the capability of maintaining a good balance between removing noise and preserving the original geometric details. Their reconstructions have clear visual appearance and high ENL, as compared with the results of RSN-ADL, which demonstrates the advantage of the proposed smoothness regularizer.

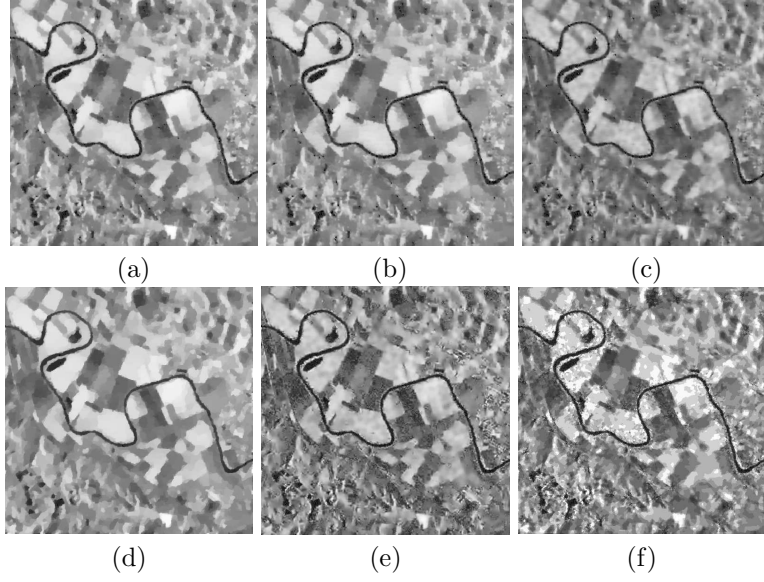


Figure 14: Results for the first SAR image. (a) MNR-ADL-SR<sub>1</sub>. (b) MNR-ADL-SR<sub>2</sub>. (c) RSN-ADL. (d) MIDAL. (e) MNR-DL-TV-2. (f) DFN.

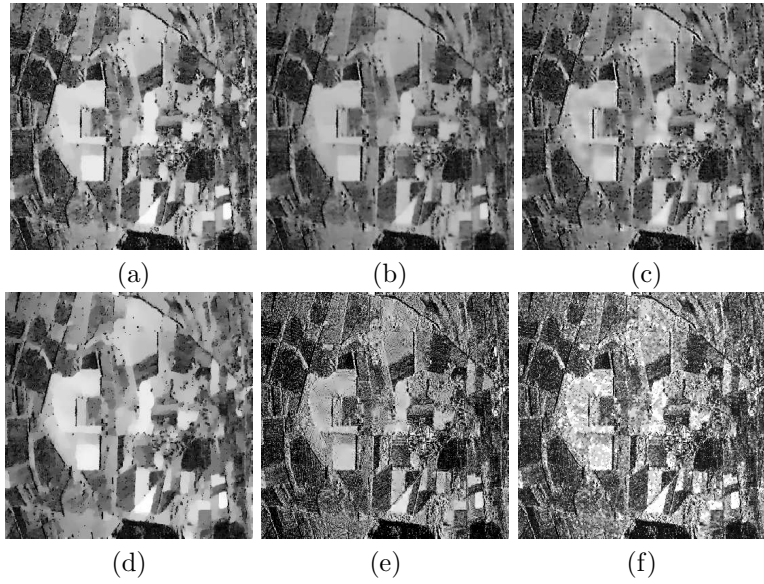


Figure 15: Results for the second SAR image. (a) MNR-ADL-SR<sub>1</sub>. (b) MNR-ADL-SR<sub>2</sub>. (c) RSN-ADL. (d) MIDAL. (e) MNR-DL-TV-2. (f) DFN.

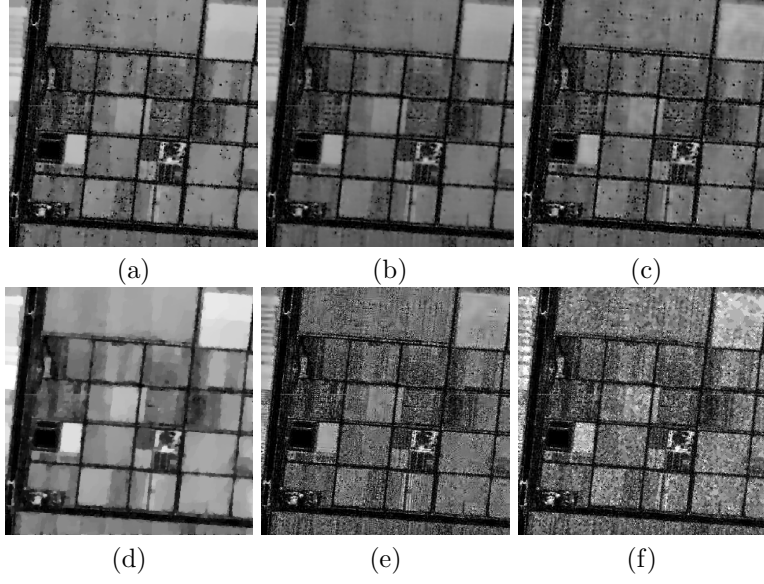


Figure 16: Results for the second SAR image. (a) MNR-ADL-SR<sub>1</sub>. (b) MNR-ADL-SR<sub>2</sub>. (c) RSN-ADL. (d) MIDAL. (e) MNR-DL-TV-2. (f) DFN.

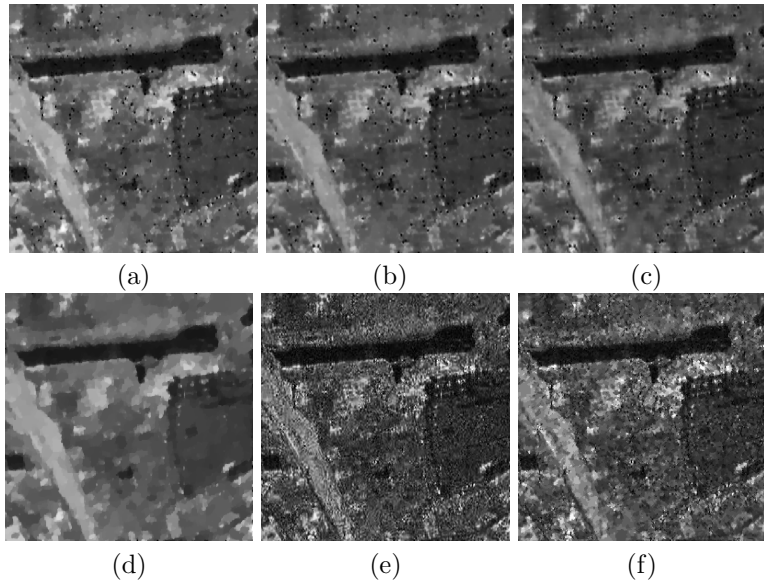


Figure 17: Results for the second SAR image. (a) MNR-ADL-SR<sub>1</sub>. (b) MNR-ADL-SR<sub>2</sub>. (c) RSN-ADL. (d) MIDAL. (e) MNR-DL-TV-2. (f) DFN.

## 435 6. Conclusion

We have proposed a new multiplicative noise removal algorithm and an optimization method corresponding to this model. The denoising task was considered in the log-domain and formulated as an objective function consisting of a data fidelity term and two regularizers. The data fidelity term was derived  
440 from the statistical property of the multiplicative noise, and the regularizers were based on a learned analysis dictionary and the pixel-wise differences of the image, respectively. In order to address the optimization for recovering the image, a variable splitting technique was applied and the ADMM framework was carefully adapted. In the update of the variable related to the smoothness  
445 regularizer, a relaxation approach was employed to convert the variables in different forms to a unified one. Simulation results with synthetic noisy images and real SAR images demonstrated the promising performance of the proposed method, especially for a relatively high noise level.

## Acknowledgement

450 The authors thank the Associate Editor and the anonymous reviewers for their contributions to improving the quality of the paper.

## References

- [1] C. Oliver, S. Quegan, Understanding Synthetic Aperture Radar Images, SciTech Publishing, 2004.
- 455 [2] S. Durand, J. Fadili, M. Nikolova, Multiplicative noise removal using l1 fidelity on frame coefficients, Journal of Mathematical Imaging and Vision 36 (3) (2010) 201–226.
- [3] J. W. Goodman, Speckle Phenomena in Optics: Theory and Applications, Roberts and Company Publishers, 2007.

- 460 [4] J. M. Bioucas-Dias, M. A. Figueiredo, Multiplicative noise removal using variable splitting and constrained optimization, *IEEE Transactions on Image Processing* 19 (7) (2010) 1720–1730.
- [5] Y.-M. Huang, L. Moisan, M. K. Ng, T. Zeng, Multiplicative noise removal via a learned dictionary, *IEEE Transactions on Image Processing* 21 (11) 465 (2012) 4534–4543.
- [6] J.-S. Lee, Digital image enhancement and noise filtering by use of local statistics, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2) (1980) 165–168.
- [7] V. S. Frost, J. A. Stiles, K. S. Shanmugan, J. C. Holtzman, A model for 470 radar images and its application to adaptive digital filtering of multiplicative noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2) (1982) 157–166.
- [8] D. T. Kuan, A. A. Sawchuk, T. C. Strand, P. Chavel, Adaptive noise smoothing filter for images with signal-dependent noise, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2) (1985) 165–177. 475
- [9] H. Xie, L. E. Pierce, F. T. Ulaby, SAR speckle reduction using wavelet denoising and markov random field modeling, *IEEE Transactions on Geoscience and Remote Sensing* 40 (10) (2002) 2196–2212.
- [10] F. Argenti, L. Alparone, Speckle removal from SAR images in the undecimated wavelet domain, *IEEE Transactions on Geoscience and Remote Sensing* 40 (11) (2002) 2363–2374. 480
- [11] G. Aubert, J.-F. Aujol, A variational approach to removing multiplicative noise, *SIAM Journal on Applied Mathematics* 68 (4) (2008) 925–946.
- [12] J. Shi, S. Osher, A nonlinear inverse scale space method for a convex multiplicative noise model, *SIAM Journal on Imaging Sciences* 1 (3) (2008) 485 294–321.

- [13] Y.-M. Huang, M. K. Ng, Y.-W. Wen, A new total variation method for multiplicative noise removal, *SIAM Journal on Imaging Sciences* 2 (1) (2009) 20–40.
- 490 [14] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena* 60 (1) (1992) 259–268.
- [15] L. Rudin, P.-L. Lions, S. Osher, Multiplicative denoising and deblurring: theory and algorithms, in: *Geometric Level Set Methods in Imaging, Vision, and Graphics*, Springer, 2003, pp. 103–119.
- 495 [16] Y. Hao, X. Feng, J. Xu, Multiplicative noise removal via sparse and redundant representations over learned dictionaries and total variation, *Signal Processing* 92 (6) (2012) 1536–1549.
- [17] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, *IEEE Transactions on*
- 500 *Signal Processing* 54 (11) (2006) 4311–4322.
- [18] S. Nam, M. E. Davies, M. Elad, R. Gribonval, The cospase analysis model and algorithms, *Applied and Computational Harmonic Analysis* 34 (1) (2013) 30–56.
- 505 [19] M. Yaghoobi, S. Nam, R. Gribonval, M. Davies, Constrained overcomplete analysis operator learning for cospase signal modelling, *IEEE Transactions on Signal Processing* 61 (9) (2013) 2341–2355.
- [20] R. Rubinstein, T. Peleg, M. Elad, Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model, *IEEE Transactions on Signal Processing*
- 510 *61* (3) (2013) 661–677.
- [21] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Z.-F. Han, J. Chambers, Analysis SimCO algorithms for sparse analysis model based dictionary learning, *IEEE Transactions on Signal Processing* 64 (2) (2016) 417–431.



- [22] J. Dong, W. Wang, J. Chambers, Removing speckle noise by analysis dictionary learning, in: Sensor Signal Processing for Defence (SSPD), 2015, pp. 7193–7197.
- [23] J. Dong, W. Wang, W. Dai, Analysis SimCO: A new algorithm for analysis dictionary learning, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 7193–7197.
- [24] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Foundations and Trends in Machine Learning* 3 (1) (2011) 1–122.
- [25] S. Ravishankar, Y. Bresler, Learning sparsifying transforms, *IEEE Transactions on Signal Processing* 61 (5) (2013) 1072–1086.
- [26] P.-A. Absil, R. Mahony, R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2009.
- [27] W. Dai, T. Xu, W. Wang, Simultaneous codeword optimization (SimCO) for dictionary update and learning, *IEEE Transactions on Signal Processing* 60 (12) (2012) 6340–6353.
- [28] D. P. Bertsekas, J. N. Tsitsiklis, *Introduction to Probability: Second Edition*, Athena Scientific, 2002.
- [29] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [30] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.
- [31] A. Chambolle, An algorithm for total variation minimization and applications, *Journal of Mathematical Imaging and Vision* 20 (1-2) (2004) 89–97.
- [32] T. F. Chan, J. J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, 2005.

- 540 [33] Y. Chen, S. Levine, M. Rao, Variable exponent, linear growth functionals  
in image restoration, *SIAM Journal on Applied Mathematics* 66 (4) (2006)  
1383–1406.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality as-  
sessment: from error visibility to structural similarity, *IEEE Transactions*  
545 *on Image Processing* 13 (4) (2004) 600–612.
- [35] F. Argenti, A. Lapini, T. Bianchi, L. Alparone, A tutorial on speckle re-  
duction in synthetic aperture radar images, *IEEE Geoscience and Remote*  
*Sensing Magazine* 1 (3) (2013) 6–35.

## Appendix

550 In this section, the reason that problem (38) can be addressed with the  
gradient descent iteration described as (39) is explained in detail.

Let the function  $m(x, y)$  denote the pixel values of the image  $\mathbf{M}_R$  for the  
pixel indices  $x, y \in \Omega$ . Similarly, the given matrix  $\mathbf{N}_R$  can be denoted as  $n(x, y)$ .  
Using these new notations, the problem (38) can be written as the functional  
minimization problem as follows

$$\arg \min_m \iint_{\Omega} \left\{ \lambda_2 \left( \left( \frac{\partial m}{\partial x} \right)^2 + \left( \frac{\partial m}{\partial y} \right)^2 \right) + \frac{\gamma_2}{2} (n - m)^2 \right\} dx \, dy. \quad (43)$$

Define

$$\begin{aligned} & F(m(x, y), m_x, m_y) \\ &= \lambda_2 \left( \left( \frac{\partial m}{\partial x} \right)^2 + \left( \frac{\partial m}{\partial y} \right)^2 \right) + \frac{\gamma_2}{2} (n - m)^2 \\ &= \lambda_2 (m_x^2 + m_y^2) + \frac{\gamma_2}{2} (n - m)^2, \end{aligned} \quad (44)$$

where  $m_x$  and  $m_y$  represent the partial derivatives  $\frac{\partial m}{\partial x}$  and  $\frac{\partial m}{\partial y}$  respectively, and

the problem (43) can be written as

$$\arg \min_m \iint_{\Omega} F(m(x, y), m_x, m_y) dx dy \quad (45)$$

The Euler-Lagrange equation associated with this problem is given by [32]

$$\frac{\partial F}{\partial m} - \frac{\partial}{\partial x} \left( \frac{\partial F}{\partial m_x} \right) - \frac{\partial}{\partial y} \left( \frac{\partial F}{\partial m_y} \right) = 0. \quad (46)$$

Since

$$\frac{\partial F}{\partial m} = \gamma_2(m - n), \quad (47)$$

$$\frac{\partial}{\partial x} \left( \frac{\partial F}{\partial m_x} \right) = 2\lambda_2 \frac{\partial m_x}{\partial x} \quad (48)$$

and

$$\frac{\partial}{\partial y} \left( \frac{\partial F}{\partial m_y} \right) = 2\lambda_2 \frac{\partial m_y}{\partial y}, \quad (49)$$

the Euler-Lagrange equation (46) is equivalent to

$$\gamma_2(m - n) - 2\lambda_2 \left( \frac{\partial m_x}{\partial x} + \frac{\partial m_y}{\partial y} \right) = 0, \quad (50)$$

which can be addressed numerically [32]. In the  $k$ -th iteration,  $m$  is updated according to the following iteration

$$m^{(k+1)} = m^{(k)} + \tau \left[ 2\lambda_2 \left( \frac{\partial m_x^{(k)}}{\partial x} + \frac{\partial m_y^{(k)}}{\partial y} \right) - \gamma_2(m^{(k)} - n) \right], \quad (51)$$

where  $\tau$  denotes the step size. Alternatively, the iteration equation above can be written in matrix form given by equation (39) as shown in Section 4.